# TDST048取扱説明書

# 安全のしおり

間違った使い方をすると、火災や感電、事故につながることがあり危険です。
また、本製品の故障にもつながります。事故や故障を防ぐために次のことを必ずお守りください。

※ 本製品の手動操作・自走プログラムを含む取扱いによる事故・故障について、弊社では一切の責任を負いかねます。

### ●公道を走行しない

当社の製品では公道（一般道）を走行する設計は行っておりません。

### ●電源の＋、－を間違わないようにする

電源コネクタへの接続が不適切であると、本製品の故障の原因となります。
電源を入れる前に、必ずご確認の上、ご使用ください。

### ●電気を通すものの上で使用しない

電気を通すものの上に直接本製品を設置するとショートするため、火災や感電の原因となり危険です。
四隅にある取り付け穴を使用し、本製品を他のものから離して設置、または電気を通さないものの上に設置してください。

### ●定格の入力電圧範囲内で使用する

定格の入力電圧範囲外で使用すると、動作不良となるだけではなく、本製品の故障につながります。
仕様範囲外の条件において使用された場合については、動作は保証できません。

### ●電源投入時にリード線や半田部などに素手で触らない

電源投入時に素手で触ると、感電の原因となります。

### ●車輪に指や衣服等を触れないようにする

巻き込まれて怪我をする可能性があり危険です。

### ●改造した場合は保証致しません

本製品の改造やソフト変更等を行った場合は、故障があっても保証致しません。

### ●バッテリーの保証は致しません

バッテリーの過放電、過充電等による破損については、保証致しません。

# 目次

各部説明

接続構成図

手動制御・自律制御 仕様書

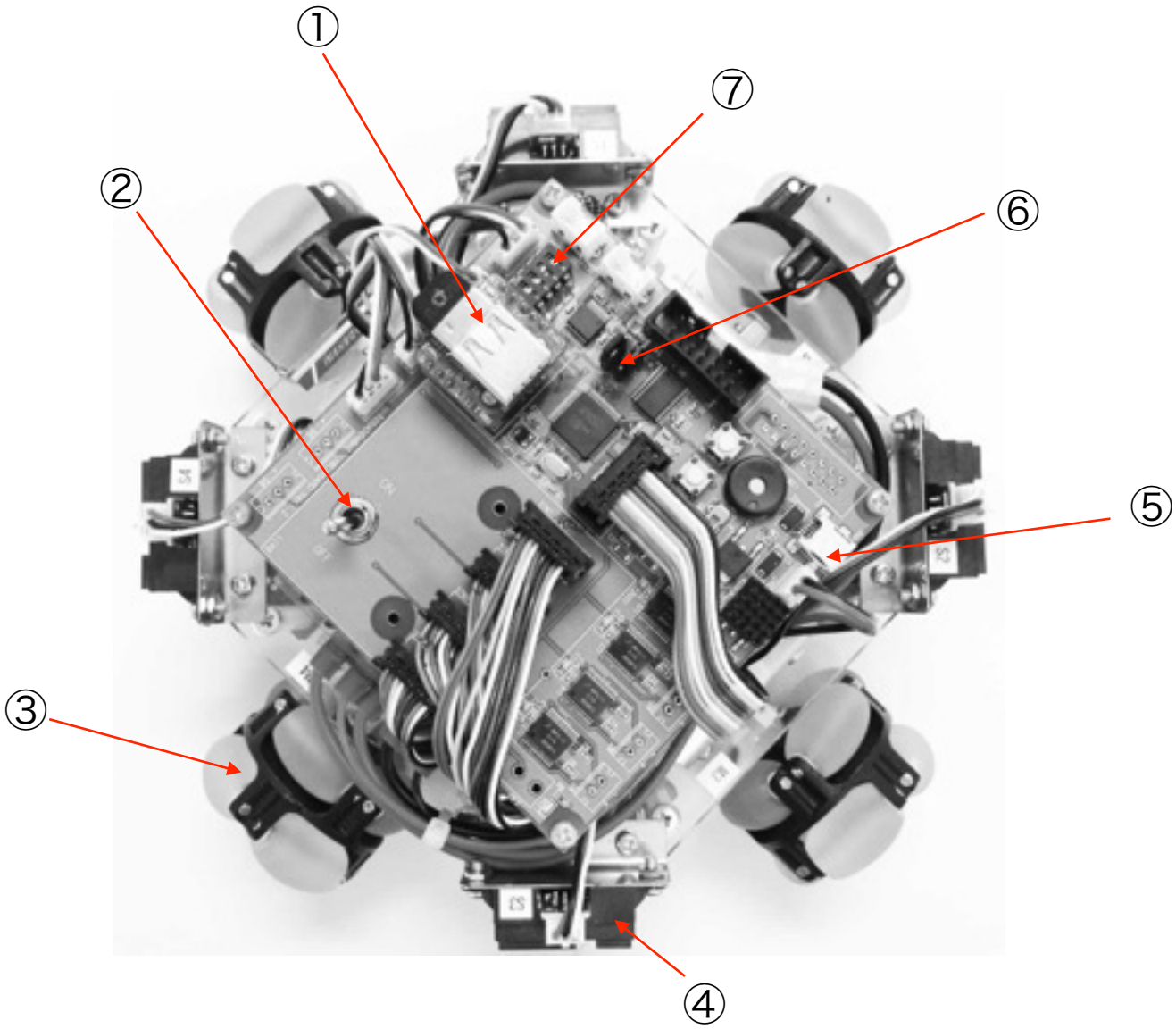
PC制御仕様書

BlueTooth通信仕様書

H８基板仕様書

TDST048メンテナンス

# 各部説明




| | | |
|---|---|---|
| ① | Bluetooth | (参考:BlueTooth通信仕様書) |
| ② | 電源スイッチ | (参考:手動制御・自律制御仕様書) |
| ③ | オムニホイール | オムニホイールTD-48 |
| ④ | センサー | シャープ測距モジュール　ＧＰ２Ｙ０Ａ２１ＹＫ |
| ⑤ | USB端子 | (参考:接続構成図) |
| ⑥ | 通信切替ジャンパー（J1） | (参考:H8制御基板仕様書) |
| ⑦ | モードスイッチ | (参考:H8制御基板仕様書) |

# 接続構成図

(参考:BlueTooth通信仕様書)

Bluetooth通信

SBDBT

(参考:手動制御・自律制御 仕様書)

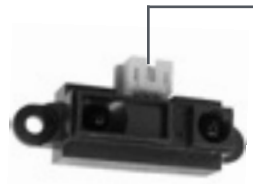
リモコン制御

VS-C3 (ロボット専用無線コントローラ「V-コントローラ」)

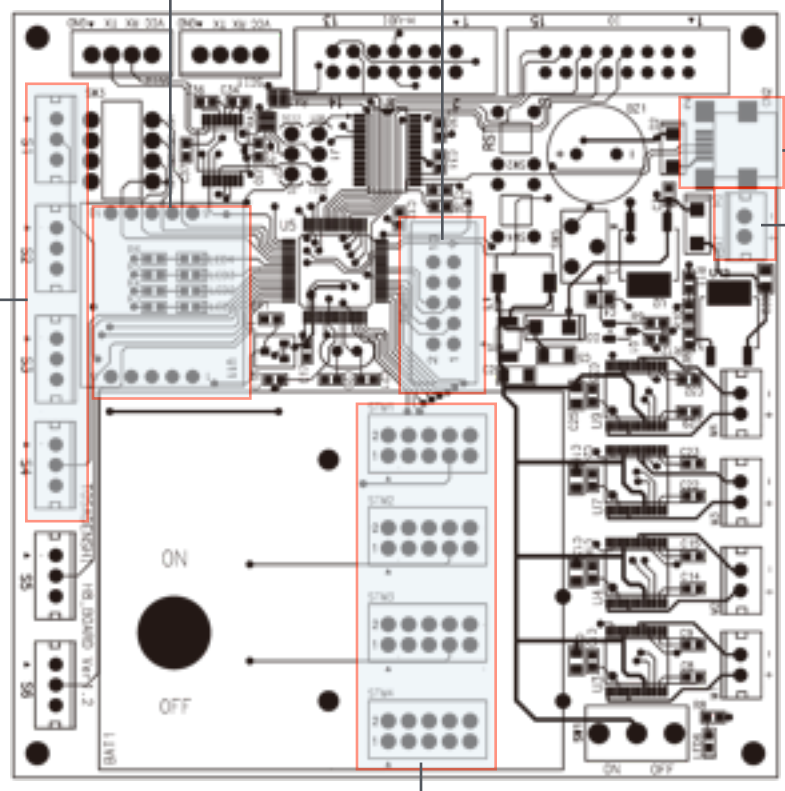
付属の変換ケーブルを接続してください。
(※)極性を間違えないように気をつけて下さい。

センサー接続

コネクタ：EHR-3(JST)
コンタクト：
SEH-001T-P0.6(JST)



シャープ測距モジュール
GP2Y0A21YK



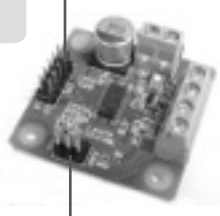

PC制御・プログラム書込み

ミニUSBケーブル

USBケーブル

(参考:PC制御仕様書)

モーター



ステッピング
モータドライバ
(L6470キット)

ステッピングモーター
(SM-42BYG011-25)

電源接続

コネクタ：EHR-2(JST)
コンタクト：
SEH-001T-P0.6(JST)

リポバッテリー EA1200 11.1V

（※）ステッピングモータドライバの詳細については、
付属CDROM内のドライバ資料をご参照下さい。

# 手動制御・自律制御
# 仕様書

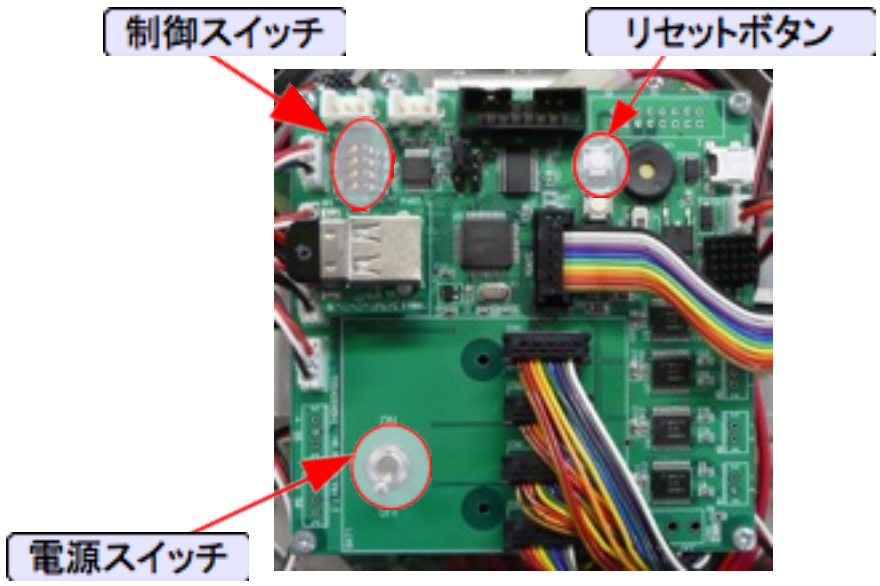
1. 手動制御モード
2. 自律制御モード

## 1.手動制御モード

（1）CPUボードの制御スイッチを全てOFFにします。

（2）電源スイッチをONにします。

（3）コントローラと受信モジュール間のペアリングを行います。（初回起動時のみ）
受信モジュールのCONNECTボタンを一度押すと赤色LEDが高速点滅を行い、
ペアリング待機モードに入ります。この状態でコントローラのANALOGボタンを押すと
ペアリングを実行します。ペアリングが完了すると、緑と赤のLEDが点灯状態になります。



（4）コントロールが可能になります。

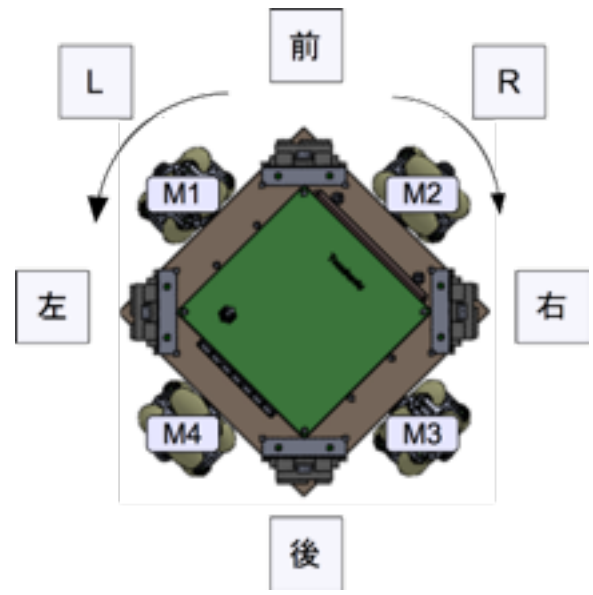
・ANALOGジョイスティックで全方向移動操作が可能です。

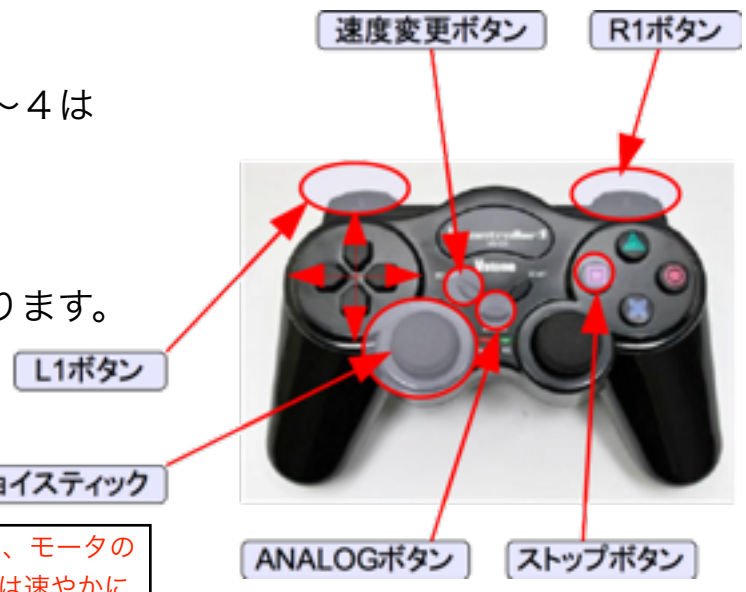
・コントローラ上部のL1及びR1ボタンで旋回動作をします。
（旋回動作時は速度設定の80%で動作します。）

・SELECTボタンでPWM変更ができます。
（初期値は40%です。ボタンを押す度にビープ音と共に
Dutyが20→40→60→80%と変更します。）

・リモコンからのコマンド指令がある場合は、LED１～４は
各モータの駆動状態に応じて点灯します。
（動作中で点灯、非動作中で消灯）

・各モータの配置と進行方向の関係は左図のようになります。




（※）バッテリー電圧が約10V以下になった場合、LED１～４が点滅し、モータの励磁を強制的に停止するようにソフト制御しています。その場合は速やかに充電するようにして下さい。

（※）バッテリー電圧は使用前、使用後と定期的にテスターで確認するようにして下さい。

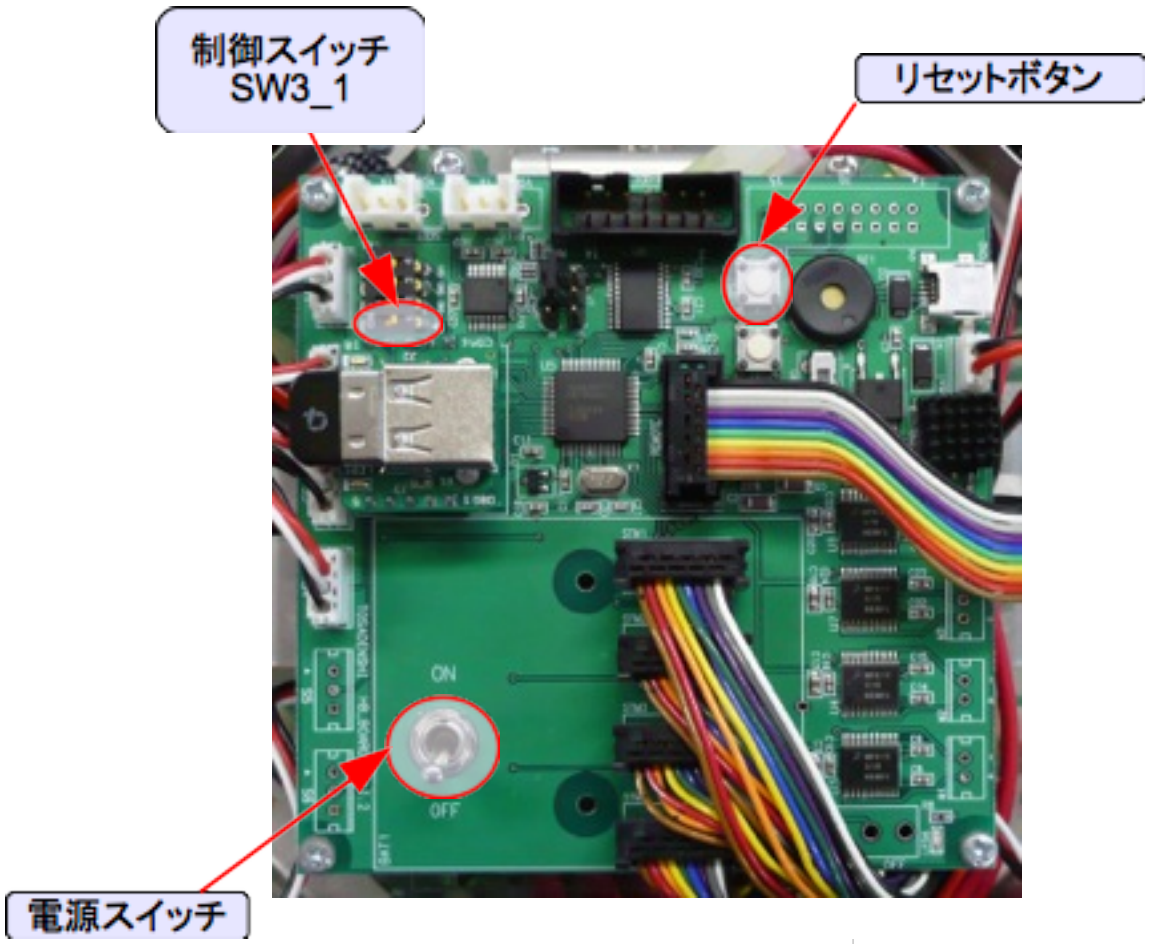
## 2.自律制御モード

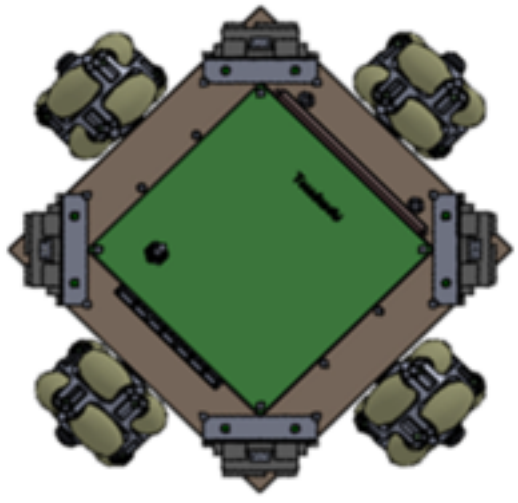
（１）CPUボードの制御スイッチのSW3_1のみONにします。

（２）電源スイッチをONにします。

（３）PSDセンサーの反応に応じて、ロボットは移動します。

※自立制御時のPWMは３０％固定での動作になります。





進行方向

PIDセンサーに手を近づけると、
それに反発する方向に移動します。
4つのセンサー全てが反応する等、
センサー反応の組み合わせに
よっては動作しない場合があります。

（※）バッテリー電圧が約10V以下になった場合、LED１～４が点滅し、モータの励磁を強制的に停止するようにソフト制御しています。その場合は速やかに充電するようにして下さい。

（※）バッテリー電圧は使用前、使用後と定期的にテスターで確認するようにして下さい。

# PC制御仕様書

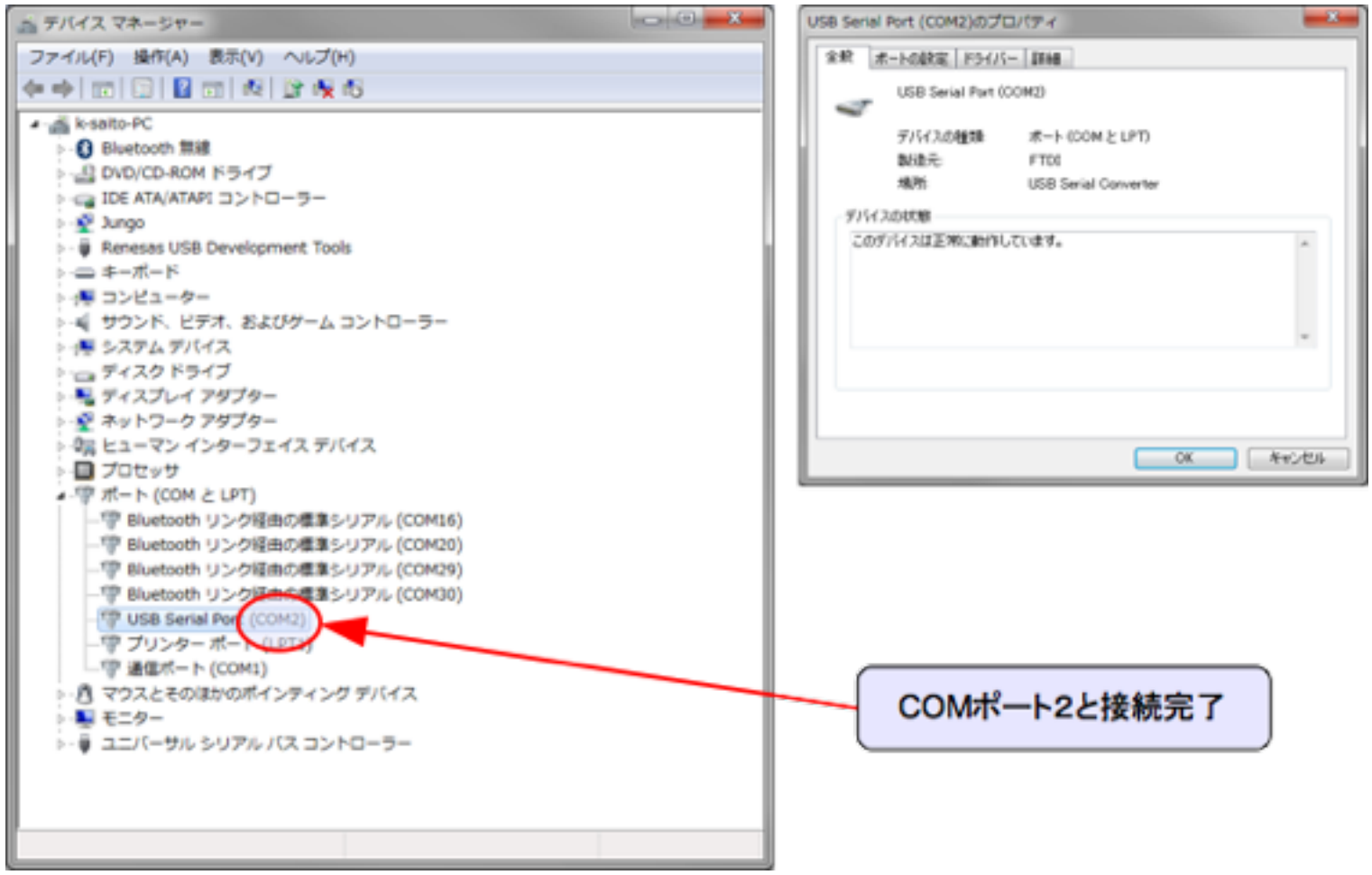

1. シリアル通信の設定
2. VB6サンプルプログラムについて
    - サンプルEXEファイルの実行について
    - サンプルEXEファイルの操作説明
    - 送信コマンド
3. DirectX SDKのインストール
4. Windows Vista/7 でのDirectX 利用方法

# 1. シリアル通信の設定

① 接続されたパソコン上ではUSBシリアル・ドライバによって仮想COMポートと認識されます。
使用する際にはドライバのインストールが必要となります。

② ミニUSBケーブルでPCと基板を接続すると、デバイスドライバのインストールが開始します。
ネット上から最適なドライバがインストールされない場合は、CDROM内の
「FTDIドライバ」フォルダを選択し、ドライバのインストールを実行してください。

③ 「マイコンピュータ」を右クリックし[プロパティ]＞[デバイスマネージャ]を開きます。
ポートを選択して、USB SerialのCOMポートが追加されていることを確認します。




CPUボードと通信を行なう際のシリアルポートの設定は下記の通りです。
通信ソフトの設定を下記の設定に合わせて下さい。

ビットレート：9600bps
データビット：8
パリティ：偶数
ストップビット：1
フロー制御：なし

# 2. VB6サンプルプログラムについて

## サンプルEXEファイルの実行について

VBサンプルプログラムは、PC用コントローラを接続して使用することが可能です。
VBサンプルプログラム上でPC用コントローラの使用を行うために、DirectXのインストールが必要となります※。
(※　本VBサンプルプログラムを実行する場合のシステム構成となります。
直接シリアル通信でコマンドを送る場合はDirectXのインストールは必要ありません。)

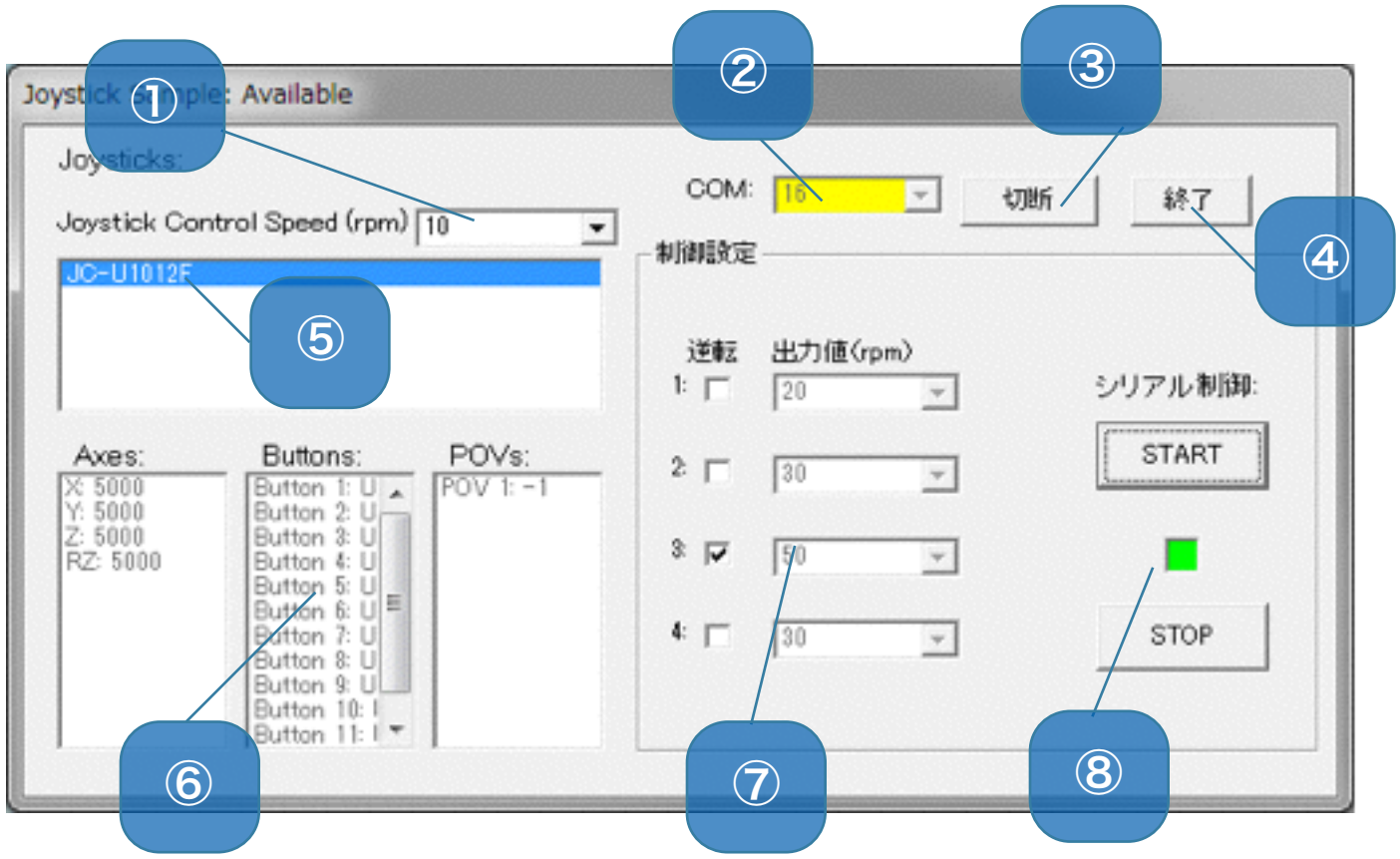
ご利用のPC用コントローラにより、ボタンの配置は異なります。
必ず、最初はロボットの車輪が接地しない状態でテストを行って下さい。

# 2. VB6サンプルプログラムについて

## サンプルEXEファイルの操作説明

CDROM内「sample_program」→「VB6_sample」→「vb_joystick.exe」で実行ファイルを起動します。

(※)Direct Xがインストールされていない場合は(5-4)のDirectX SDKの導入を実行してください。

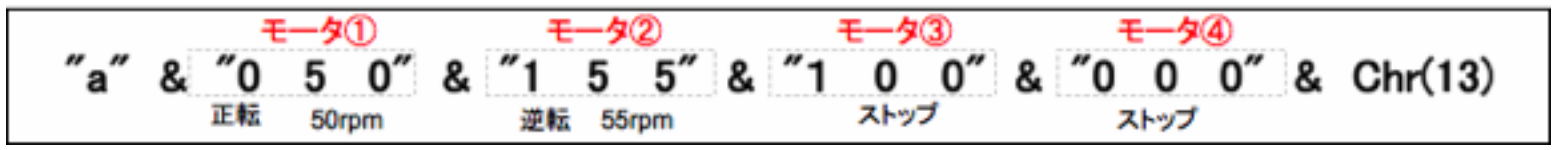



① コントローラ使用時の速度設定
② 接続する通信ポートを選択
③ 通信ポートへの接続及び切断ボタン
④ プログラムの終了ボタン
⑤ 使用するコントローラを選択（使用する場合は必ずクリックしてください。）
⑥ コントローラの操作値を表示
⑦ モーター個別制御の速度・回転方向を選択
⑧ モーター個別制御の開始及び停止ボタン（個別制御中は緑が点灯）※

※ジョイスティックをＰＣに接続されていない場合でも、個別にシリアル制御可能です。
※シリアル制御のSTARTを押した後は、STOPを押すまでジョイスティックでの操作は受け付けません。

## 送信コマンド

サンプルプログラムからCPU基板に実際に送信するAPIコマンドは以下のようなフォーマットになります。サンプルを利用せず、直接シリアルコマンドを送信する場合や独自のソフトウェアをご利用になられる場合も同様のコマンドを送信することで制御可能です。




- 最初の文字が制御設定(0：正転、 1：逆転)、後2文字が速度設定（99rpm以下)となります。
- コマンドの開始と終了判定の為、最初に「a」と最後に改行コード(0x0d)を送信します。

### VBサンプルプログラム内の送信コマンド生成ブロック例

```
'### /// 送信するコマンド モータ順（①②③④）

If (lstButton.List(0) = "Button 1: Down") Then
  MSComm1.Output = "a" & "200" & "200" & "200" & "200" & Chr(13) '手動強制ストップ
  Move_flg = 0 '//--- 姿勢判定フラグを初期化

ElseIf (lstButton.List(4) = "Button 5: Down") Then '//— 左回転
    If (Move_flg = 2 Or Move_flg = 0) Then
        MSComm1.Output = "a" & "1" & mspeed & "1" & mspeed & "1" & mspeed & "1" & mspeed & Chr(13)
        Move_flg = 2
    Else
      Stop_Motion
    End If

ElseIf (lstButton.List(5) = "Button 6: Down") Then '//--- 右回転
    If (Move_flg = 1 Or Move_flg = 0) Then
        MSComm1.Output = "a" & "0" & mspeed & "0" & mspeed & "0" & mspeed & "0" & mspeed & Chr(13)
        Move_flg = 1
    Else
      Stop_Motion
    End If

ElseIf (lstHat.List(0) = "POV 1: 0") Then '//--- デジタル前進
    If (Move_flg = 3 Or Move_flg = 0) Then
        MSComm1.Output = "a" & "0" & mspeed & "1" & mspeed & "1" & mspeed & "0" & mspeed & Chr(13)
        Move_flg = 3
    Else
      Stop_Motion
    End If
```

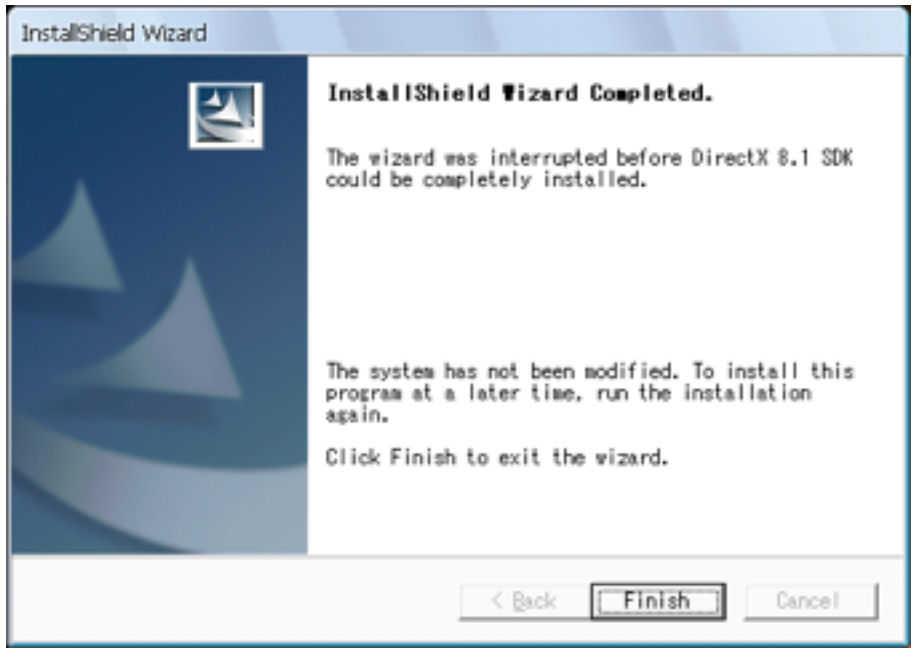
# 3. DirectX SDKのインストール

VBサンプルプログラムはDirectX8を利用しているため、実行・編集ともにDirectX8のインストールが必要となります。

（※）VBサンプルプログラムの動作保証はWindowsXPの環境のみとなります。

① 付属CDROM内の「DX8.1bSDK」のフォルダを開きます。
② インストールプログラム「install.exe」を実行します。

③ 「Install DirectX 8.1 SDK」をクリックします。
④ インストーラが起動するので「Next >」をクリックしてインストールを進めてください。
⑤ 「Finish」を選択してインストール完了となります。


InstallShield Wizard
InstallShield Wizard Completed.
The wizard was interrupted before DirectX 8.1 SDK could be completely installed.
The system has not been modified. To install this program at a later time, run the installation again.
Click Finish to exit the wizard.
< Back
Finish
Cancel

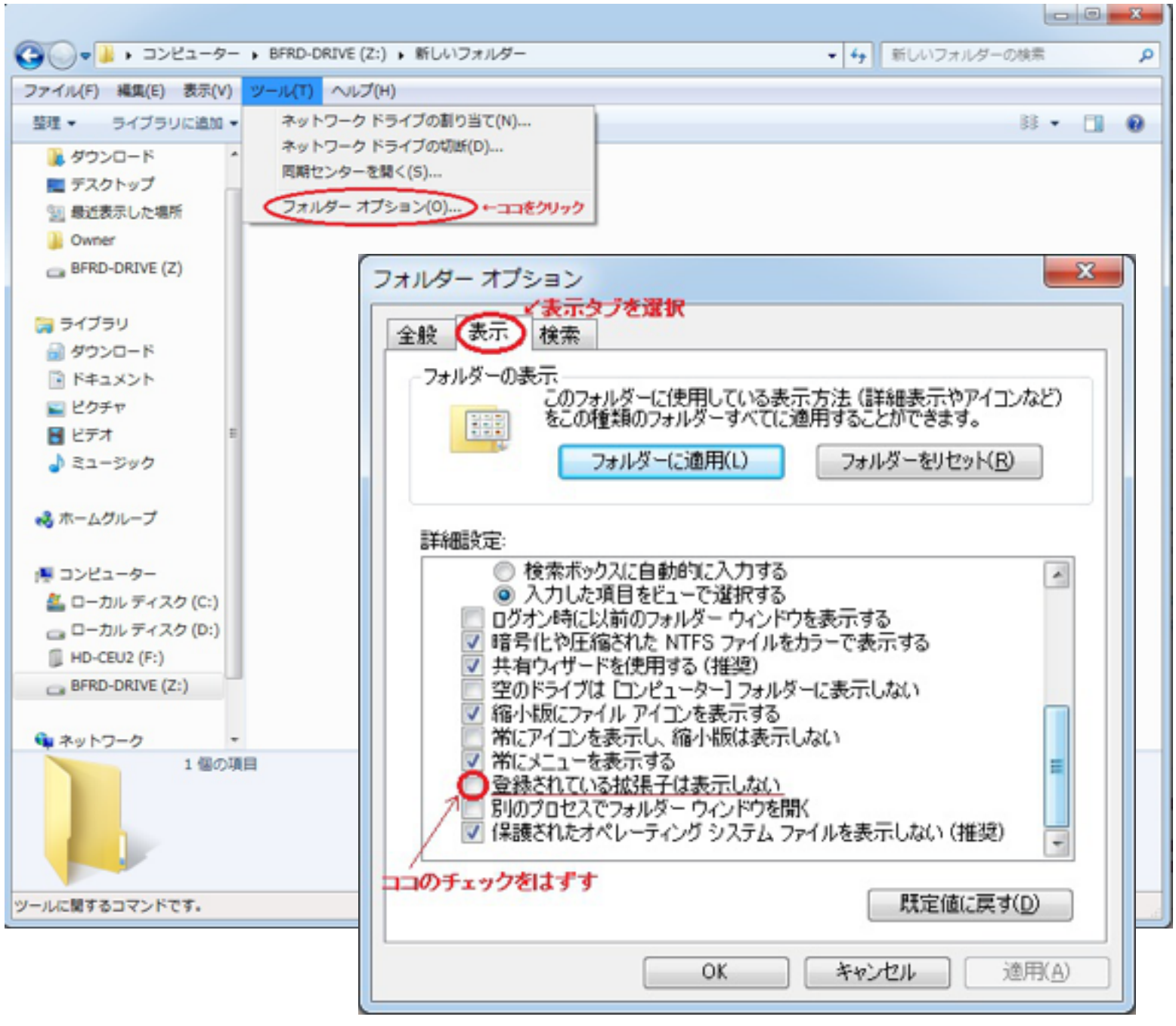
# 4. Windows Vista/7 でのDirectX 利用方法

付属のDirectXインストール後、VBサンプルプログラム上でPC用コントローラが利用できない場合にご参考ください。

※ 本作業はレジストリの登録（編集）を行います。
※ 本作業及び、作業ミスにより発生した不具合につきましては自己責任にてご対応をお願い致します。
※ 管理者権限での作業が必要となります。

① 下記URLよりDirectX 8.1b Runtime for Windows 2000 (日本語版) をダウンロードする。
http://www.microsoft.com/ja-jp/download/details.aspx?id=10968

② フォルダオプションで「拡張子の表示」を行う設定に変更する。



③　先ほどダウンロードしたファイルの拡張子を「***.exe」から「***.cab」に変更する。

④　「***.cab」を解凍ソフトで解凍する。

⑤　解凍されたファイルの中から「dx7vb.dll」と「dx8vb.dll」を探し出しコピーする。
　　コピー先は
　　**32bitOS**の場合　「C:\Windows\System32」
　　**64bitOS**の場合　「C:\Windows\SysWOW64」

⑥　「管理者権限」でコマンドプロンプトを起動し、下記のコマンドを入力する。

　　**32bitOS**の場合
　　C:\Windows\System32\regsvr32.exe dx7vb.dll
　　C:\Windows\System32\regsvr32.exe dx8vb.dll

　　**64bitOS**の場合
　　C:\Windows\SysWOW64\regsvr32.exe dx7vb.dll
　　C:\Windows\SysWOW64\regsvr32.exe dx8vb.dll

（※　上記作業によりWindowsの動作が不安定になった場合は下記コマンドによりレジストリへの登録を解除することができます。）

例：64bitOSのレジストリからdx8vb.dllを解除する場合
　　C:\Windows\SysWOW64\regsvr32.exe /u dx8vb.dll

SBDBTモジュールを利用

# BlueTooth通信仕様書

## iPhone/iPad (iOS)で通信する

iPhone/iPadアプリの使い方

# iPhone/iPad (iOS)で通信する

**PC等、BLE通信以外でBlueToothをご利用の場合は、
事前にSBDBTのファームウェア書換えが必要となります。
購入後に書換えが必要となった場合は、
下記ホームページを参考にファームウェアの書き換えを
行うか、お問い合せにて対応させていただきます。**

ランニングエレクトロニクス様ホームページ
http://runningele.web.fc2.com

**iPhone / iPad用アプリケーションはベータ版でありAppStore等で配布しておりません。
添付CDROM内　iPhoneアプリサンプルより、Xcodeのソースファイルをコンパイルしてご使用ください。**

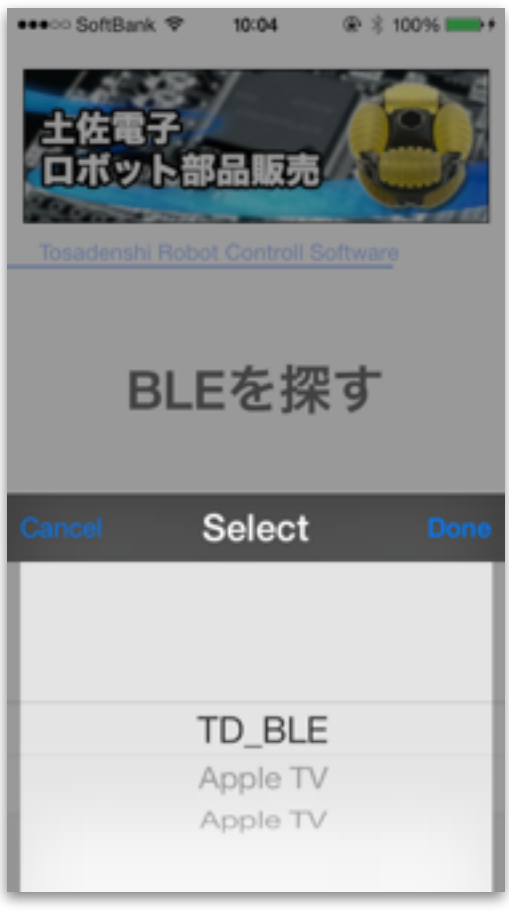
当サンプルプログラムは
SBBLE (Konashi) をベースに作成しております。
http://sbble.micutil.com/#id0

動作環境
・BLE対応のデバイス（iPhone4S以降）
・iOS6以降のOS(iOS7推奨)

## iPhone / iPadアプリの使い方

① 「TDBLE」アプリケーションを起動すると、左図の画面が表示されます。
ロボット本体（BlueToothモジュール）の電源をONにして、「BLEを探す」をタップして下さい。

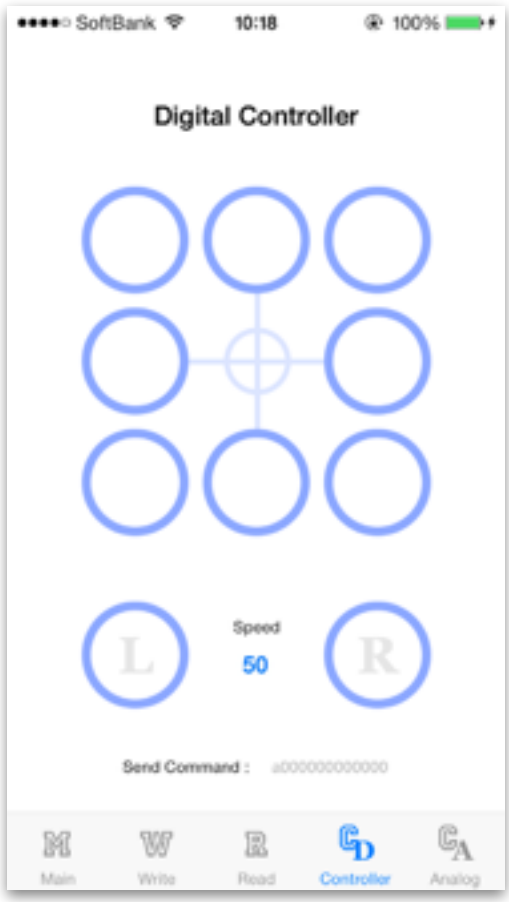
（※　本アプリケーションは、BlueToothを利用します。
iPhone/iPad本体の設定で、BlueToothが利用出来るようにしてください。）




② デバイス名の選択画面が出ますので「TD_BLE」を選択して「Done」をタップして下さい。

③　画面下のタブでモードの切替を行えます。
「Write」タブは、１列分の文字列をデバイスに送信します。
中心のテキストフィールドに文字列を記載し、送信ボタンを
タップして下さい。

④　「Read」タブは、モジュールからの受信を表示します。
受信文字数に対し通信速度が早い場合は、文字化けやアプリ
ケーションの強制終了が発生する可能性があります。




⑤ 「Controller」タブは、
土佐電子製ロボットのコントロールを行います。
８方向の円及び、L・Rをタップすることでコントロールが可能です。
速度の変更はSpeed部分の数字をタップして下さい。

⑥ 「Analog」タブは、
土佐電子製ロボットのコントロールを行います。
タップした場所を起点として、指を動かした方向に指を動かした距離分の速度で移動を行います（ソフトウェアアナログパッド）。
回転は下部のバーをL・Rに動かすことで行います。

# H8制御基板 仕様書

## 各部説明

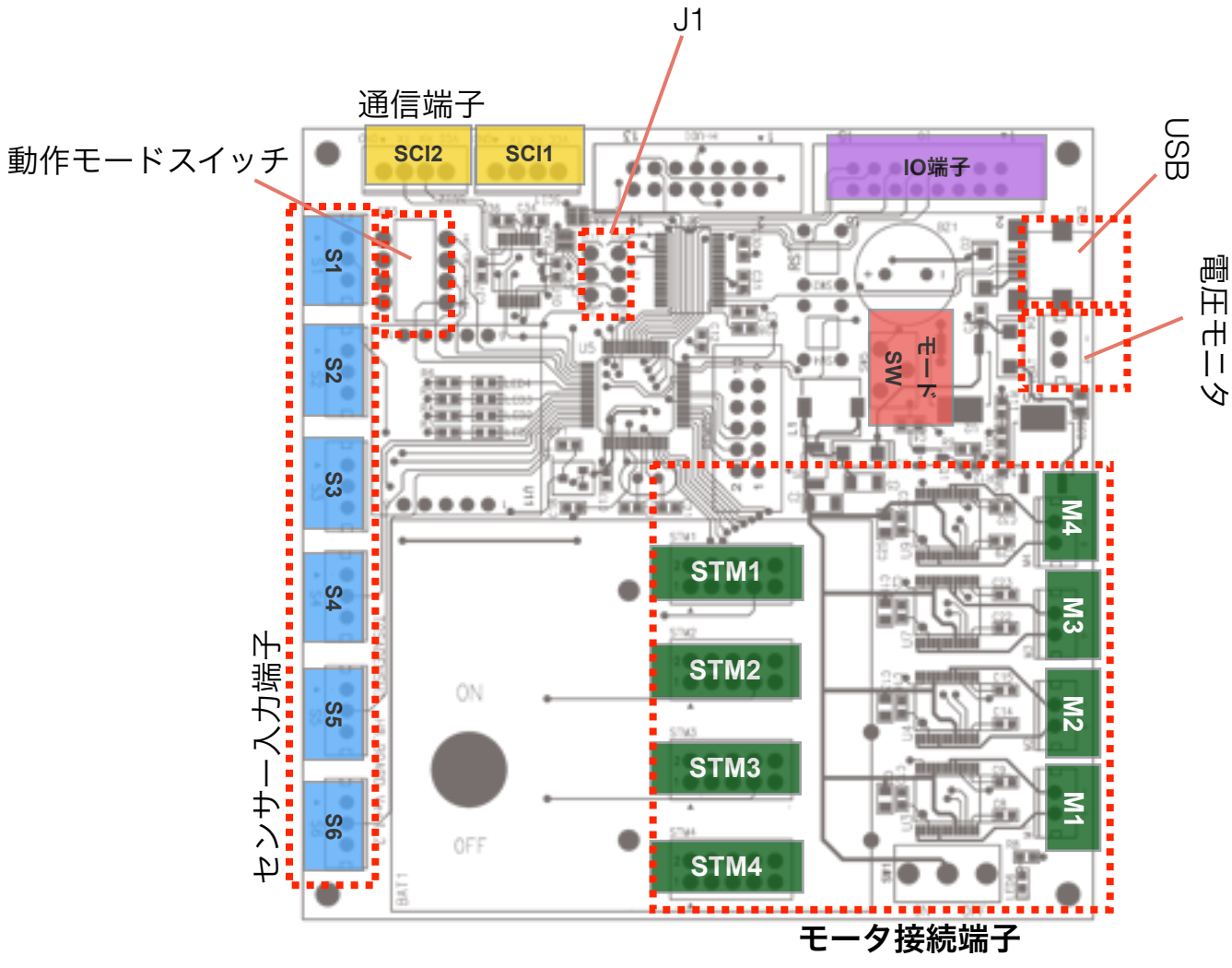
## 開発の準備

1. コンパイラ、ROM書込みソフトインストール
2. サンプルプログラムのコンパイル
3. プログラムの書込み

## サンプルプログラム

主要ライブラリ

# 各部説明




## ●センサー入力端子（S1～S6）

| | |
|---|---|
| 1 | 5v |
| 2 | 信号 |
| 3 | GND |

## ●動作モードスイッチ

| | |
|---|---|
| 1 | SW1 |
| 2 | SW2 |
| 3 | SW3 |
| 4 | NMI |

## ●モータ制御端子（DCモータ用：M1～M4）

| | |
|---|---|
| 1 | M+ |
| 2 | M- |

３ｖ駆動のDCモータを推奨

※STM１～４はTDSTO専用のステッピングモータ用となります。

## ●モードSW

| | | |
|---|---|---|
| 1 | L | ロード |
| 2 | W | ライト |

ソフトウェアを書込はWを選択

## ●通信端子（SCI1,SCI２）

| | |
|---|---|
| 1 | GND |
| 2 | TX |
| 3 | RX |
| 4 | VCC 3.3v |

J1で切替

SCI1・USB

SCI2・BT(BlueTooth)

SCI1はプログラム書込みと兼用となります。
J1でSCI1を選択するとUSBからプログラムの
書込みができなくなります。
ステッピングモータをご利用される場合はSCI1・USB
からテキスト出力が不可能となります。

## ●IO端子（7ch）

| | | | |
|---|---|---|---|
| 1 | 0 | 2 | GND |
| 3 | 1 | 4 | GND |
| 5 | 2 | 6 | GND |
| 7 | 3 | 8 | GND |
| 9 | 4 | 10 | GND |
| 11 | 5 | 12 | GND |
| 13 | 6 | 14 | GND |
| 15 | 7 | 16 | GND |

## ●USB端子

ソフトウェアの書込みを行います

# 開発の準備

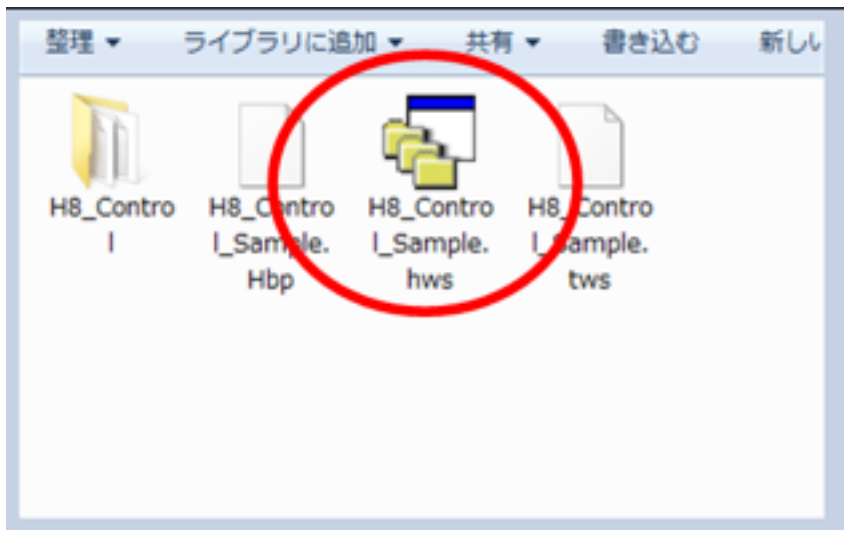
## 1．コンパイラ、ROM書込みソフトインストール

- C言語で開発する際に使用可能なルネサステクノロジより無料配布されている
  ”High-performance Embedded Workshop(評価版）”を用いて、サンプルプログラムを
  書き込む手順について解説したものです。
  HEWの仕様及びインストールにつきましてはルネサスホームページ
  http://japan.renesas.com/products/tools/ide/hew/hew_mid_level_landing.jsp　をご覧下さい。

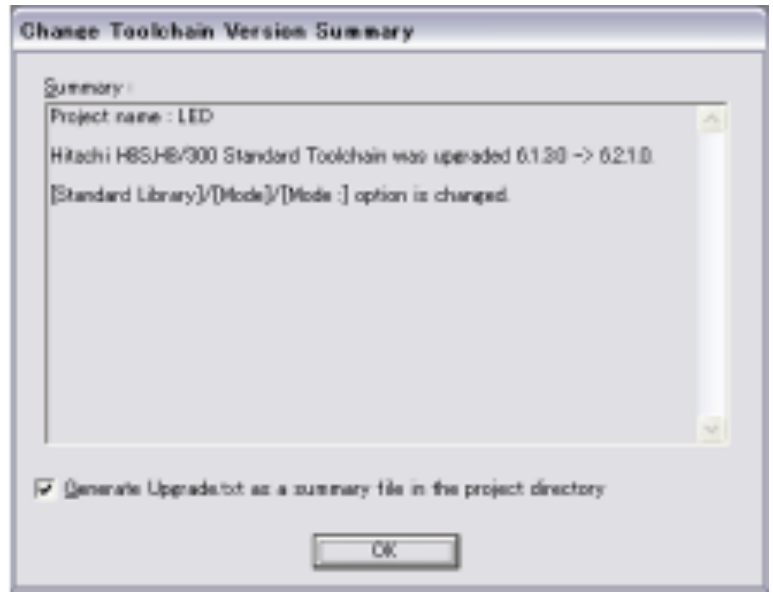
- CDROM内の開発環境フォルダにある”h8v7000_ev.exe”を実行することで、
  HEWの開発環境をインストールすることができます。また、”fdtv408r01.exe”を実行することで、
  後述のプログラム書き込みソフト（FDT)をインストールすることができます。

## 2．サンプルプログラムのコンパイル

- HEWの起動とサンプルプログラムの読み込み
  （１）サンプルプロジェクト[H8_sample_program]を[c:\WorkSpace]にコピーします。
  （※）OSがインストールされているドライブ名がCの場合。
  （WorkSpaceフォルダはHEWインストール時に自動で生成されます。）
  （２）Windowsの[スタートメニュー]から[プログラム]>[Renesas]>[High-performanceEmbedded Workshop]>[High-performance Embedded Workshop]を起動します。
  （※）起動時に”ようこそ”画面が表示されますが、キャンセルして下さい。
  （３）メニューの[ファイル]から[ワークスペースを開く]を選択し、サンプルプロジェクト内にある[H8_Control_Sample.hws]を開きます。

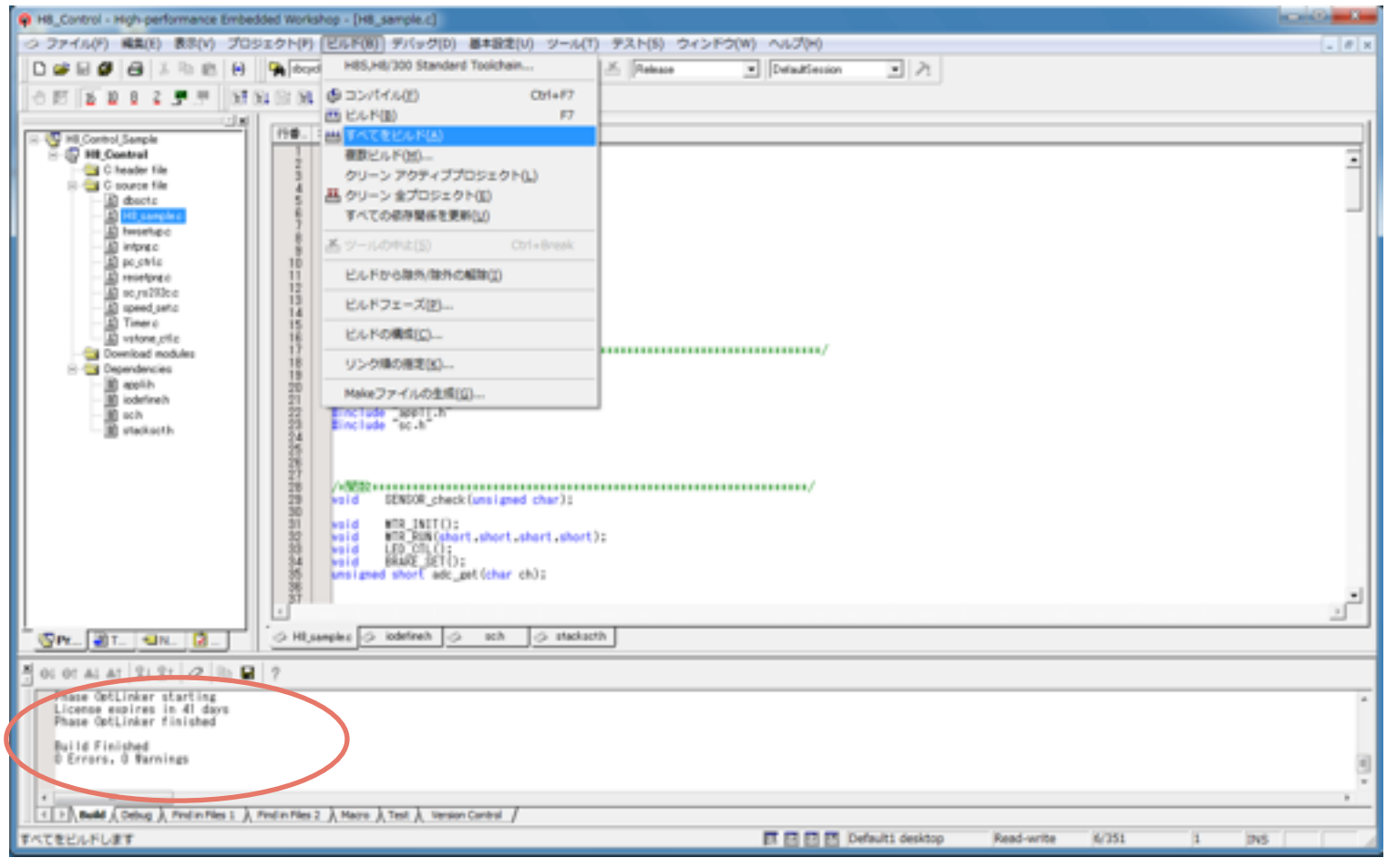


（※）プロジェクトを開く際に、ツールチェインが異なる場合があり、その場合、変更を求められることがあります。変更する場合は、全て[OK]を押すことで変更できます。

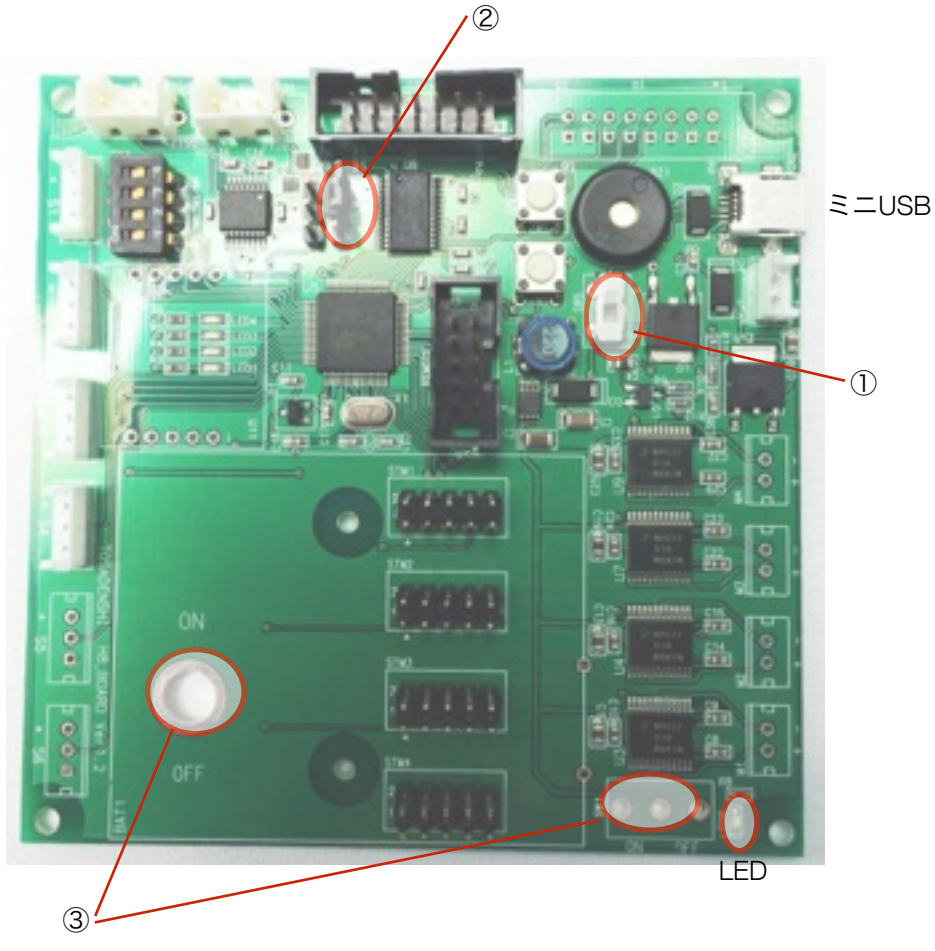



（４）main関数があるH8_sample.cファイル等を編集し、所望のプログラムを作成して下さい。
編集が終わると、[ビルド]→[すべてをビルド]を実行することで、コンパイルを実行することができます。

（※）初回コンパイル時やフォルダ変更等した場合は、Warningが発生することがあります。
2回目以降は発生しなくなりますので、問題ありません。



# 3．プログラムの書込み




（1）電源OFFの状態で、制御基板をプログラミング
モードに変更します。
※TDSTOとTDOシリーズでは③電源SWの場所が異なります

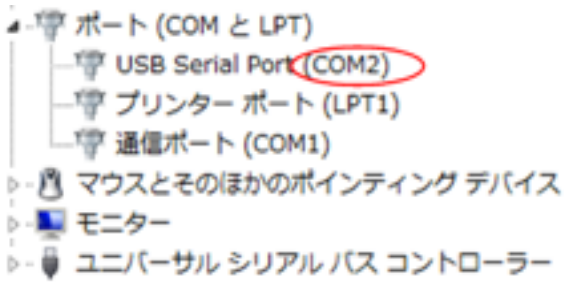
①スライドスイッチをWに設定
②ジャンパーをUSBモードに設定（デフォルの状態）
※変更する際は、必ず電源OFF状態で行ってください。

（2）書き込みを行うパソコンを起動します。
※USBケーブル接続時にFTDIドライバをインストール
する必要があります。
付属CDROM内のFTDIドライバを適当なフォルダに
解凍して、ドライバのインストールを行って下さい。

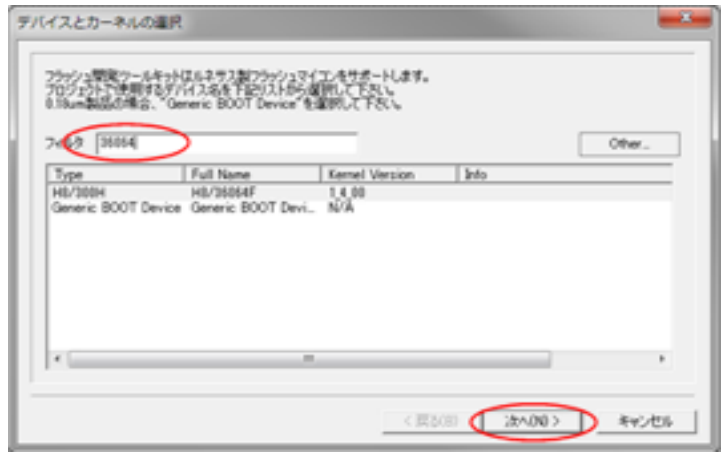
（3）制御基板とパソコンをミニUSBケーブルで接続します。

（4）接続と同時に基板のLEDが点灯します。
※パソコンのUSBから基板の電源が供給され基板の
メイン電源がONとなります。

（5）利用しているポートを確認する。
windowsのデバイスマネージャーを立ち上げ、
USB Serial PortのCOMポート番号を確認します。

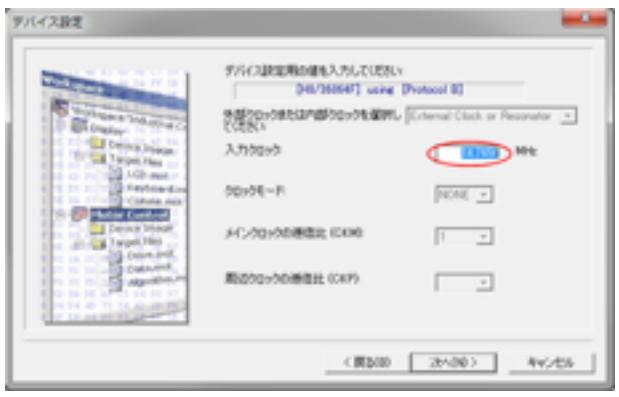



（6）スタートメニューの[プログラム]>[Renesas]>
[Flash Development Toolkit ***]>[Flash
Development Toolkit ***Basic]を起動します。
(***はバージョン番号）初回起動時には”デバイスと
カーネルの選択”という画面が発生します。
フィルタの空欄部分に ”36064”と入力すると、
”H8/36064F”が出てきますので、それを選択して
次へ進んで下さい。
（※）2回目以降は下図のような画面が起動します。
上記設定をやり直す場合は、[オプション]>[新規設定]
を選択して下さい。



（7）通信ポートの設定を行います。（5）で確認した
COMポートを選択して、次へ進んで下さい。

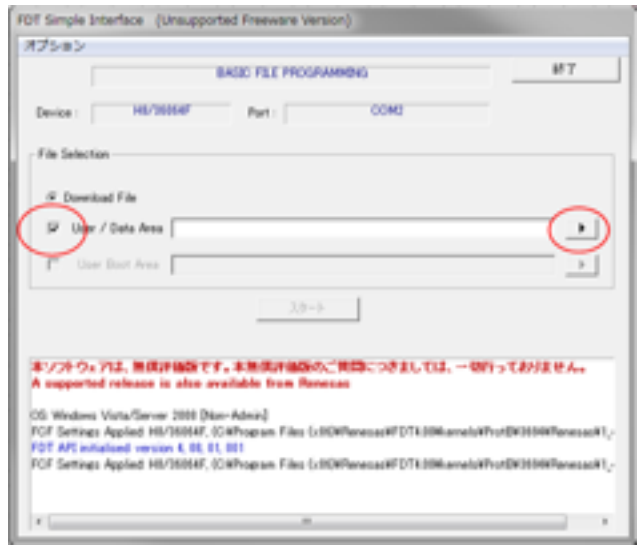
（8）デバイスの設定を行います。ここでは入力クロック
を”14.7456”に設定して次へ進んで下さい。



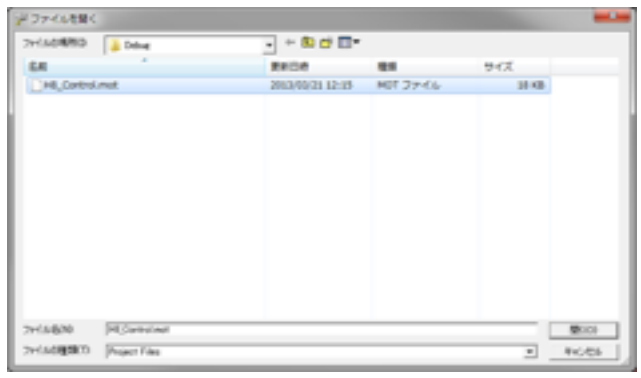
（9）接続タイプの設定を行います。 ここは何もせずに、
次へ進んで下さい。

（10） 書き込みオプションの設定を行います。
ここは何もせずに、完了を押下してください。

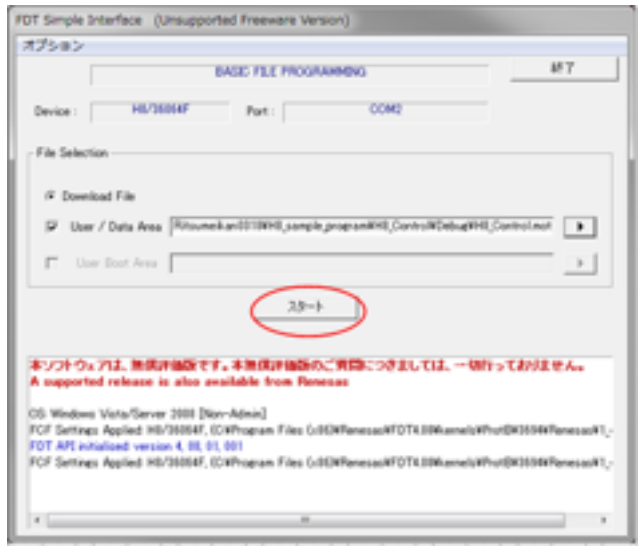
（11）設定が完了したら、[User/DataArea]を
チェックし、右向きの三角ボタンを押して、
書き込みファイルを選択します。



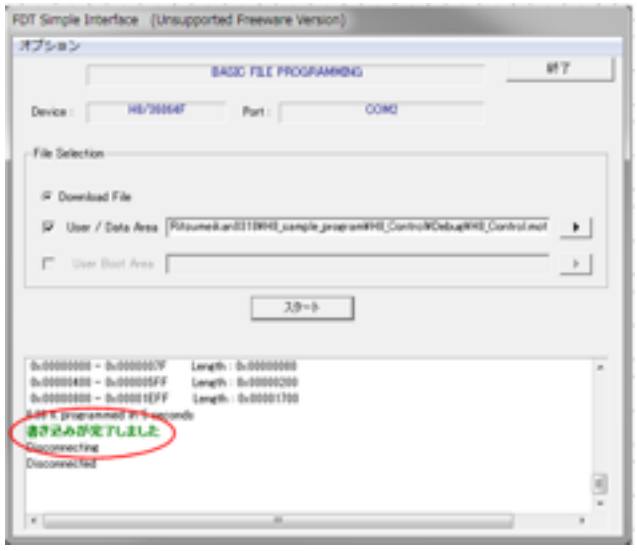
（12）書き込みファイルはプロジェクトフォルダ内の
[Debug]フォルダに生成されます。拡張子は
motとなります。



（13）書き込みファイルを選択後、スタートボタンを
押下し書き込みを開始します。



（14）無事に書き込みが完了すると、右図のような
画面が出現し、書き込みが終了します。



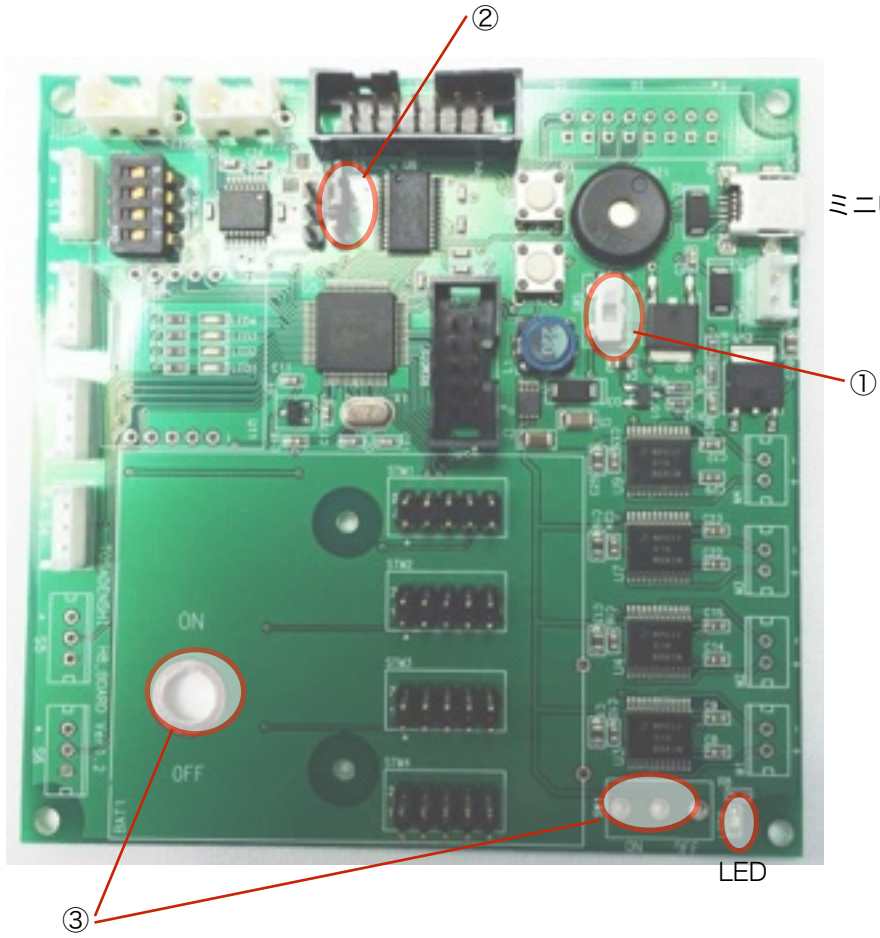

（※）ブートの合わせ込みに失敗等が出た場合は、
まず電源を入れ直して最初からリトライして
みて下さい。それでもうまくいかない場合は、
接続COMポートをできる限り若い番号のポートに
変更してリトライしてみてください

## 4．プログラムのロード

（1）制御基板からミニUSBケーブルを抜き、
電源OFFの状態にします。引抜くとLEDが消えます。

（2）①スライドSWをLに変更します。

（3）③電源起動すると、書込んだ
プログラムが実行されます。



# サンプルプログラム

付属のサンプルプログラムは大きく以下のファイルから構成されます。

・H8_sample.c　メイン関数を含むプログラムのメインファイル
・hwsetup.c　　CPUの初期化関数ファイル
・intprg.c　割込み処理記述ファイル
・pc_ctrl.c 通信系コマンド制御に関するファイル
・sc_rs232c.c　　通信関連の各処理をまとめたファイル
・Timer.c　Wait関数等のタイマー処理をまとめたファイル

グローバル変数やIOポートを定義したファイルとして以下のものがあります。

・appli.h　　グローバル変数やマクロを定義したファイル
・iodefine.h　　IOポートの定義ファイル
・sc.h　　通信系パラメータの定義ファイル
（※）上記ファイルは基本的なファイルであり、これ以外にもファイルは存在します

| 関数名 | 機能 |
|---|---|
| HardwareSetup() | CPU初期化関数 |
| set_imask_ccr(n) | n=1で割込禁止、n=0で割込許可 |
| Wait(n) | n[ms]ウェイト関数 |
| BuzzerStart() | ブザー音発振開始 |
| BuzzerStop() | ブザー音発振停止 |
| MTR_INIT() | モータ関係パラメータ等の初期化関数 |
| ap_select_SCI1() | SCIチャンネル１からのコマンド受信処理関数 |
| ap_select_SCI2() | SCIチャンネル２からのコマンド受信処理関数 |
| updatePAD | リモコンからのコマンドデータ受信処理関数 |
| getPAD(PAD_B1) | 受信コマンドからPAD_B1に割り当てられているデータを格納 |
| getPAD(PAD_B2) | 受信コマンドからPAD_B2に割り当てられているデータを格納 |
| BRAKE_SET() | ブレーキ処理関数 |
| pc_ctrl() | SCI系コマンド実行処理関数 |
| MTR_RUN(*,*,*,*) | 各DCモータへのDuty設定（-100～100%）及び駆動制御を行う。 |
| MTR_SPD(*,*,*,*) | 各ステッピングモータのRPM設定（-100～100%）及び駆動制御を行う。 |
| LED_CTL() | 動作状況に応じてLEDの点灯制御を行う関数 |
| SENSOR_check(n) | 各センサーの出力モニターを行い、センサーが反応する電圧閾値を設定する。 |

# 主要ライブラリ

## ●センサー入力S1～S6

adc_get(n)

nは１～６chで指定する。A/Dに接続された数値を返します。

使用例：AD1chに入った数値をsci2(Bluetooth)からシリアルで出力する。

```
char  temp[50];

sprintf(temp, "ADC1=%5u, ", adc_get(1));
sc2_rs232c_puts((unsigned char*)temp);
```

## ●モードスイッチ

SWn
nは１～3のモードスイッチになります。

使用例：SW１の状態でif文を構成する。SW１がONでブザーが鳴り、OFFでブザーが止まる

```
if(SW1 == 1){
        BuzzerStart();
}
else if(SW1 == 0){
        BuzzerStop();
}
```

## ●通信端子（SCI2,Bluetooth）

シリアル通信

使用例：SCI2に[TDO Start]と出力する。

```
char  temp[50];

sprintf(temp,"\rTDO Start!!\r”);
sc2_rs232c_puts((unsigned char*)temp);              //Bluetooth系
```

## ●IO端子（7ch)

IO_input(n)
nは０～７ｃｈまでを指定する。各IOの状況１、０の数値を返します。

使用例：IO_input(0)の状態でif文を構成する。IO_input(0)がHightでブザーが鳴り、Lowでブザーが止まる

```
if(IO_input(0) == 1){
        BuzzerStart();
}
else if(IO_input(0) == 0){
        BuzzerStop();
}
```

## ●DCモータ制御端子（M1～M４）

各モータへのDuty設定（-100～100%）及び駆動制御を行う。

使用例：DCモータM1,M2を正回転５０％で駆動し、M2,M3を逆回転６０％で駆動する。

```
short spd[4];
spd[1]=50;
spd[2]=50;
spd[3]=-60;
spd[4]=-60;
MTR_RUN(spd[1], spd[2], spd[3], spd[4]);
```

## ●ステッピングモータ制御端子（STM1～STM４）

各モータへのDuty設定（-100～100%）及び駆動制御を行う。

使用例：ステッピングモータSTM1,STM2を正回転５０％で駆動し、STM3,STM4を逆回転６０％で駆動する。

```
short spd[4];
spd[1]=50;
spd[2]=50;
spd[3]=-60;
spd[4]=-60;
MTR_SPD(spd[1], spd[2], spd[3], spd[4]);
```

# TDST048
# メンテナンス

## 1. バッテリー充電

# 1. バッテリー充電

## ❑充電方法

① 本機の電源をOFFにしてください。
② バッテリーを本機から取り外して、付属の専用充電器にて充電を行ってください。

ご利用前に必ずバッテリーと充電器の取扱説明書をお読み下さい。

- バッテリー充電時は必ずバッテリーを本機から取り外して行って下さい。
- 絶対にバッテリーを過充電させないで下さい。（専用充電器をご使用下さい。）
- バッテリーを満充電状態での保管は行わないで下さい。
- バッテリーを機器に接続したまま保管しないで下さい。
- バッテリー電圧は使用前、使用後と定期的にテスター等で確認して下さい。
- バッテリー電圧が約10V以下になった場合、LED1～４が点滅し、モータ励磁を強制的に停止するようにソフト制御しております。但し、消費電流は０にはなりませんので、ご注意下さい。
- バッテリー電圧が規定電圧以下になると、充電不可能となる場合があります。電圧低下した場合は速やかに充電を行って下さい。
- バッテリーに関する不具合やトラブルに関して、弊社は一切責任を負いかねます。